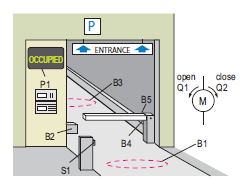
****



**PLC Applications**

Module 1: Control task planning and implementation





PREPARED BY

**Academic Services Unit**

December 2011

© Applied Technology High Schools, 2011

Module 1: Control task planning and implementation

**Module Objectives**

Upon successful completion of this module, students will be able to:

* Analyze a control task by defining its Inputs and Outputs, and its technical requirements.
* Configure control task hardware by selecting the proper LOGO! Basic module and its expansion modules.
* Develop control routines and design an Industrial Process control program from beginning to end using LOGO Soft Comfort software.
* Use normally open and normally close sinking and sourcing sensors
* Understand and apply interlocks, and latches.

**Module Contents:**

|  |  |  |
| --- | --- | --- |
|  | **Topic** | **Page No.** |
| 1.1 | Introduction | 3 |
| 1.2 | Implementation sequence | 4 |
| 1.3 | Sinking and sourcing sensors | 10 |
| 1.4 | latches using Ladder diagram | 12 |
| 1.5 | Interlocks | 13 |
| 1.6 | Text messages | 16 |
| 1.7 | Practical tasks | 17 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1.1 Introduction**  Small control tasks can be solved using LOGO! with a minimum of hardware. As you studied in PLC course input signals supply LOGO! with information on the current state of the process and any operator commands. The control relays react to these input signals in accordance with a defined program. It then generates output signals which influence the process in the intended manner via actuators (final control elements).  A simple control task can be as follow:  Customer orders are assembled on pallets as shown in figure 1.1.  Orders ready for dispatch are transported on a conveyor system to the truck ramp. The two keys S2 and S3 permit transportation of the pallets. The pallets are only transported further if at least one of the start keys is kept pressed.   |  | | --- | |  | | **Figure 1.1 Conveyor system** |   The pallets are transported to the end position where they activate the limit contact B4 ("limit switch"). B4 prevents a pallet from being unintentionally transported beyond the end position and thus falling off. The master switch S1 suppresses all movements of the chain conveyor drive. All inputs are connected to 220 V.  **1.2 Implementation sequence**  What is the procedure for planning and implementing a control task?  The implementation sequence is shown in Fig 1.2   |  | | --- | |  | | **Figure 1.2**  **Implementation sequence** |   The first step the **task analysis** is extremely important, at this stage the following steps are to be followed:   * Define all inputs and outputs. * Assign a variable name for each input and output. * Analyze the system properties and its technical requirements and conditions * Use the technical requirements to describe the relation between the outputs and inputs, at this stage Boolean expression can be used.   The inputs and outputs of the conveyor system mentioned before are listed below:   |  |  | | --- | --- | | **Inputs** | | | Main control switch | S1 | | Start pushbutton 1 | S2 (N.O) | | Start pushbutton 2 | S3 (N.O) | | Limit switch | B4 (N.C) | | **Output/s** | | | Motor | M1 |   **Table 1.1 I/O for conveyor system**  The number and type of input (N.O or N.C) and output (active high or active low) objects are based on the technical requirements and technical specifications of the control task. Table 1.2 shows the technical requirements for the conveyor system   |  |  |  | | --- | --- | --- | | **Output /s** | **ON-requirements** | **OFF-requirements** | | Motor (M1) | Main switch (S1) is ON **and** at least one of the start keys is pressed (S2 **or** S3) **and** limit switch (B4) is **not** pressed | Main switch (S1) is OFF **or**  Limit switch (B4) is pressed |   **Table 1.2 Requirements for conveyor system**  You must have noticed that in the ON side we **negate** the OFF requirements, in this example pressing the limit switch (B4) is under OFF requirements, to shift this under the ON requirements it becomes B4 **not** pressed which is in boolean. Since the limit switch is already normally close then it becomes 4.  Now to write the logic expression or the Boolean equation that describes the relation between inputs and outputs replace each **AND** by (**.**) and each **OR** by (**+**)  The second stage is **hardware configuration**, at this stage a device list can be used to assist selection of the LOGO! Controller and expansion modules, In table 2.1 it is recorded if the object has to be connected to an input (DI = Digital Input) or to an output (DO = Digital Output), in our example 4 digital inputs and one digital output is required. As mentioned in the control task description all inputs are to be connected to 220 V that means our selection is 115/240 V LOGO! Basic module and there is no need for any expansion module.   |  | | --- | | untitled.bmp | | **Table 1.3 LOGO! Basic module selection** |   (This table is used to select the proper LOGO! Basic module you do not need to memories this table)  The following expansion modules can be connected to the LOGO! In case the control task requires more inputs or outputs either digital or analog.   |  | | --- | | untitled.bmp | | **Table 1.4 Expansion modules** |   ( This table is used when there is a need to select expansion module(s), you do not need to memories this table)  After selecting the proper LOGO! an assignment list should be created, in this list all input and output objects are addressed and assigned to LOGO! inputs and outputs as shown in table 1.5. A logical assignment of inputs and outputs is necessary for both the installation (hardware connection) and for generation of the program.   |  |  | | --- | --- | | **Inputs** | | | Input | Assigned address | | S1 | I1 | | S2 (N.O) | I2 | | S3 (N.O) | I3 | | B4 (N.C) | I4 | | **Output/s** | | | Output | Assigned address | | M1 | Q1 |   **Table 1.5 Assignment list**  The third stage is **software configuration**, before the control program can be developed; the project data should be entered in the "Properties" window, which can be displayed in the pull-down menu "File".   |  |  | | --- | --- | | untitled.bmp | untitled.bmp | | **Figure 1.3**  **Properties window** | |   It is very helpful to produce a connection table (Figure 1.4) but it integrates the assignment list in the project, and names are assigned to the objects which are much more appropriate during generation and checking of the program than the addresses themselves.   |  | | --- | | untitled.bmp | | **Figure 1.4 Connection table** |     The connection table is opened by selecting the menu "Edit" and then "Input/Output Names".  In order to program the LOGO! the program is initially developed according to the control task as a draft on paper. However, the program can be produced directly using the LOGO! Soft Comfort software on a PC/notebook. Correct functioning of the program can then be directly checked per simulation. Any errors can thus be corrected in advance.  While simulation it is very important to simulate the inputs properly by selecting the correct option. In our example starting keys S2 and S3 are make pushbuttons (N.O PBs) while limit switch B4 is break pushbutton, see Figure 1.5.   |  |  | | --- | --- | | untitled.bmp | untitled.bmp | | **Figure 1.5 simulation** | |   The relationship between hardware and software is coordinated during the last stage which is **commissioning**, and the system is optimized in that faults are eliminated.  many engineers write software without taking the time or effort to design it. This often comes from previous experience with programming where a program was written, and then debugged.  Lab activity 1 page: 17  **1.3 Sinking and sourcing sensors**  Sinking and sourcing terminology applies only to DC input and output circuits. Input and output points that are sinking or sourcing can conduct current in one direction only.  **Sinking sensors** allow current to flow into the sensor to the voltage common, while **sourcing sensors** allow current to flow out of the sensor from a positive source.  When discussing sourcing and sinking we are referring to the output of the sensor that is acting like a switch. In fact the output of the sensor is normally a transistor, which will act like a switch (with some voltage loss). A PNP transistor is used for the sourcing output, and an NPN transistor is used for the sinking input.  The **sinking sensor** responds to a physical phenomenon. If the sensor is inactive (nothing detected) then the active line is low and the transistor is off, this is like an open switch. When the sensor is active, it will make the active line high. This will turn on the transistor, and effectively close the switch.   |  | | --- | |  | | **Figure 1.6 Sinking sensor** |   **Sourcing sensors** are the complement to sinking sensors. The sourcing sensors use a PNP transistor, as shown in Figure 1.7. When the sensor is inactive the active line stays at the V+ value, and the transistor stays switched off. When the sensor becomes active the active line will become 0V, and the transistor will allow current to flow out of the sensor.   |  | | --- | |  | | **Figure 1.7 Sourcing sensor** |   It is important to realize that NpN and PnP are completely different to normally open and normally closed. You can have an NpN n/o or an NpN n/c, and a PnP n/o or PnP n/c sensor.  Wiring is a major concern with PLC applications, so to reduce the total number of wires; two wire sensors have become popular.  A two wire sensor can be used as either a sourcing or sinking input. In both of arrangements the sensor will require a small amount of current to power the sensor, called the **leakage current**    Lab activity 2 page: 21  **1.4 latches using Ladder diagram**  In some applications, we need to use the transient close/open buttons for the start and stop of equipment. To maintain its continuous action, latched circuits are needed. A latch is like a **sticky switch**, when pushed it will turn ON, but stick in place; it must be pulled to turn OFF. A latch in ladder logic uses one contact to latch, and a second to unlatch.  Figure 1.8 shows 2 different latches used to control a motor.   |  | | --- | |  | | **Figure 1.8 Start and stop a Motor** |   For safety reasons it is preferred to use a normally close pushbutton to stop a Motor.  Lab activity 3 page: 22  **1.5 Interlocks**   |  |  | | --- | --- | | Have you ever tried to insert a coin inside a vending machine and to order more than one item? The machine gives you only one item, because it uses interlocks.  **Interlocks** are used to ensure two incompatible events cannot occur at the same time. | **untitled.bmp** | | **Figure 1.9 Vending machine** |   **Interlocking** is holding a system operation until certain conditions are met. These are often required for safety on industrial equipment to protect workers. a good example is in **reversible motor control**, where two motor contactors are wired to switch polarity (or phase sequence) to an electric motor, and we don't want the forward and reverse contactors energized at the same time.   |  |  | | --- | --- | | • For the protection of persons, interlocking must be hardware-based.  • For the protection of machines and products, interlocks can be hardware based and/or software-based. | untitled.bmp | | **Figure 1.10 Hardware interlock** |   **Software-based interlock**  Back to our reversible Motor control task, the task is as follows:  Two normally open pushbuttons are used to turn 12 VDC motor ON; one will run the motor in the forward direction and the other in the reverse direction. A normally close pushbutton is used to turn the motor OFF.  We will start by analyzing the task:   |  |  | | --- | --- | | **Inputs** | | | Input | Assigned address | | PB1 (N.O) | I1 | | PB2 (N.O) | I2 | | PB3 (N.C) | I3 | | **Output/s** | | | Output | Assigned address | | Forward direction | Q1 | | Reverse direction | Q2 |   **Table 1.6 Assignment list, reversible Motor**   |  |  | | --- | --- | | Forward direction requirements | Reverse direction requirements | | PB1 is pressed **AND** PB3 is **NOT** pressed and the motor is **NOT** running in the reverse direction. | PB2 is pressed **AND** PB3 is **NOT** pressed and the motor is **NOT** running in the forward direction. |   **Table 1.7 System requirements**  Since we have two outputs we need two Boolean expressions:  Figure 1.11 shows the ladder diagram for reversible Motor control task; latches are used in this diagram to maintain continuous forward and reverse actions, while interlocks are used to ensure forward and reverse actions cannot occur at the same time.   |  | | --- | | untitled.bmp | | **Figure 1.11 LAD for reversible Motor control task** |   Lab activity 4 page: 23  **1.6 Text messages**  “Message texts” programming block is used to show text messages on the LOGO! display unit, Input **En** triggers the output and shows the message while **P** is the message priority.  Acknowledgement disabled (Ack = Off):  The message text is hidden with a 0 to 1 signal transition at input En.  Acknowledgement enabled (Ack = On):  After input En is reset to 0, the message text is displayed until acknowledged by pressing the OK button. The message text cannot be acknowledged as long as input En is high.     |  |  | | --- | --- | | **Connection** | **Description** | | Input **En** | A 0 to 1 transition at En (Enable) triggers the output of the message text. | | Input **P** | P is the priority of the message text.  0 is the lowest, 30 the highest priority.  Quit: Acknowledgement of the message text | | Parameter | **Text:** Input of the message text  **Par:** Parameter or actual value of another, already configured function (see "Visible parameters or actual values")  **Time:** Shows the continuously updated time-of-day  **Date:** Shows the continuously updated date  **EnTime:**Shows the time of the 0 to 1 transition  **EnDate:**Shows the 0 to 1 transition of the date | | Output **Q** | Q remains set as long as the message text is queued. |   **Table 1.8 Message texts block**  Lab activity 5 page: 24 |

|  |
| --- |
| **1.7 Practical tasks**  **Lab activity 1 : Sorting Machine** |
| **Objective:** Analyze a control task, develop and implement a practical solution for a control task. |
| A conveyor belt starts and stops using green and red pushbuttons respectively, green light goes ON to indicate that conveyor belt is moving. If any object is detected by an inductive sensor white light goes ON until the object is sorted out by a branching arm and detected by a fiber optic barrier. The conveyor belt stops if any of the following occurs:   * + The stop red pushbutton is pressed.   + The number of the metallic objects detected is 4   + The optical sensor at the end of the conveyor belt detects 5 non-metallic objects.   **Use The Edutrainer prototype production line to implement this control task.**   1. Create an I/O assignment list.  |  |  | | --- | --- | | **Inputs** | | | Input | Address | | Green pushbutton (N.O) | I1 | | Red pushbutton (N.C) | I2 | | Inductive sensor (N.O) | I12 | | Fiber optic barrier | I11 | | Optical sensor | I10 | | **Outputs** | | | Output | Address | | Green Indicator Light | Q1 | | White Indicator light | Q2 | | Branching arm | Q7 | | Conveyor belt motor | Q8 |  1. Analyze the system requirements and write the Boolean expression for each output.   For RS latch block use the expression form given in figure 1.12   |  | | --- | | untitled.bmp  Note: | | **Figure 1.12 RS latch Boolean expression** |  |  |  |  | | --- | --- | --- | | **System requirements** | | | | Output | Requirements | Boolean expression | | Green indicator | Conveyor belt is ON |  | | White indicator | Branching arm is ON |  | | Branching arm | It goes ON when I12 detects metal and it keeps ON until I11 detects a sliding object (metallic object is sorted out) |  | | Conveyor belt | It goes on when I1 is pressed and keeps ON till  I2 is pressed  Counter 1 counts 4  Counter 2 counts 5 | C1, C2 refer to counters 1 and 2 |  1. Draw the FBD for this control task.   Note : pressing the green pushbutton should reset all counters   |  | | --- | | untitled.bmp |  1. Use the LOGO! Soft comfort software to solve this task.   While solving this control task you are required to produce a connection table and to describe the task in the properties window.   1. Simulate the program and fill in the table provided below by writing either **ON** or **OFF**:  |  |  |  |  |  | | --- | --- | --- | --- | --- | | Conditions:  Green PB is pressed and then released  White PB is not pressed  All sensors are inactive | | | | | | Output | Conveyor belt | Branching arm | Green indicator | White indicator | | Status | ON | OFF | ON | OFF | | Conditions:  Green PB is pressed and then released  White PB is not pressed  Only inductive sensor is active | | | | | | Output | Conveyor belt | Branching arm | Green indicator | White indicator | | Status | ON | ON | ON | ON | | Conditions:  Green PB is pressed and then released  White PB is not pressed  All sensors are inactive , 4 metallic objects are sorted out | | | | | | Output | Conveyor belt | Branching arm | Green indicator | White indicator | | Status | OFF | OFF | OFF | OFF |  1. Run and test the program. |
| **Lab activity 2 : N.O and N.C sinking and sourcing sensors** | |
| **Objective:** Use normally open and normally close sinking and sourcing sensors   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Some sensors are 4 wire sensors, in general wires are as follows:   |  |  | | --- | --- | | Blue | 0 VDC | | Brown | +V | | White | N.C | | Black | N.O | | photo1.JPG | | | **Figure 1.14 4 wire sensor** | | | 1. Connect the fiber optic barrier sensor as shown in figure 1.15. 2. Create and test the programs given   In the table provided below and write your comments.   |  |  | | --- | --- | | Program | Comments | | untitled.JPG | When activating the sensor light goes ON | | untitled.JPG | When activating the sensor light goes OFF | | | photo.JPG | | | **Figure 1.15** | | | 1. Connect the fiber optic barrier sensor as shown in figure 1.16. 2. Create and test the programs given   In the table provided below and write your comments.   |  |  | | --- | --- | | Program | Comments | | untitled.JPG | When activating the sensor light goes OFF | | untitled.JPG | When activating the sensor light goes ON | | | photo.JPG | | | **Figure 1.16** | | | |

|  |
| --- |
| **Lab activity 3: Latches using ladder diagram** |
| **Objective:** understand and apply latches.  Try the following control routines, and write down your comments:   |  |  |  | | --- | --- | --- | | **Ladder diagram** | **Actions** | **Comments** | | untitled.JPG | Press and release the green PB. | Green light stays ON as long as the green PB is pressed. | | untitled.JPG | Press and release the green PB. | Green light goes ON once the green PB is pressed but it can’t be switched OFF. | | untitled.JPG | Press and release the green PB.  Then press the red one. | Green light goes ON once the green PB is pressed but it can’t be switched OFF. (I2 is useless) | | untitled.JPG | Press and release the green PB.  Then press the red one. | Green light goes ON once the green PB is pressed and it goes OFF once the red PB is pressed | | untitled.JPG | Press and release the green PB.  Then press the red one. | Green light goes ON once the green PB is pressed and it goes OFF once the red PB is pressed | |

|  |
| --- |
| **Lab activity 4: Interlocks** |
| **Objective:** understand and apply Interlocks.  Use the Edutrainer kit and LOGO! soft comfort software to create and implement the following control task:  Green PB and white PB are used to move the Edutrainer table in the forward and backward directions respectively while the red one is used to stop it, table keeps moving after releasing green or white PB till it reaches any of the proximity switches, once the table is detected by a proximity switch it stops automatically.  Software- based interlock must be used to ensure that forward and backward cannot occur at the same time.   1. Draw the ladder diagram for the previous control task.  |  | | --- | | untitled.JPG |  1. Run and test your program. |

|  |
| --- |
| **Lab activity 5: Message texts** |
| **Objective:** use message texts block.  Repeat the previous task and use the “Message texts“ programming block to show “Moving Forward” when table is moving in the forward direction and “Moving backward” when table is moving in the backward direction and “Stop mode” when table is not moving.   1. Draw the ladder diagram for the previous control task.  |  | | --- | | untitled.JPG |  1. Run and test your program. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Module Exercise**  **Car park with an access control system.**  Two barriers are each equipped with a reversible motor (Q1 and Q3 open, Q2 and Q4 close), a pillar with a key switch for the entrance, and the following sensors (NC contacts):  • B1, B6: induction loops in front of the barriers  • B2, B7: reflex sensors for detection of objects "underneath" the barriers  • B3, B8: induction loops behind the barriers  The barriers are equipped with end contacts (NO contacts) for detection of the bottom (B4, B9) and top end positions (B5, B10). In addition, the entrance has a display P1 "Occupied".  If a vehicle approaches, the barrier opens if the following conditions are satisfied:   |  |  | | --- | --- | | Entrance | • Key switch S1 has been activated and  • Induction loop B1 has a "1" signal. | | Exit | • Induction loop B6 has a "1" signal. |   The barriers open up to the end position and cannot be closed as long as the reflex sensors B2 and B7 detect an object ("1" status). The areas  are closed by activation of the induction loops B3 and B8.  The signal lamp "Occupied" signals that all spaces are full. The entrance is then blocked.   |  |  | | --- | --- | | **untitled.bmp** | untitled.bmp | | **Figure 1.17 Car park with an access control system** | |  1. Create an I/O assignment list. 2. Analyze the system requirements and draw the FBD for this control task. 3. Simulate and test your program using the LOGO! Soft comfort software. |

|  |  |  |
| --- | --- | --- |
| **Simple Vending Machine.**  In a simple vending machine when you put in a Dirham, coin detector generates a “high” signal, this allows you to select one item out of 4 different items all are of the same price (1 Dirham) , 4 pushbuttons are used to order these items; one for each item.  When you press any button to get something (say, juice can), the PLC turns on the motor connected to the spiral that hold the can. The spiral spins and drops the can. 4 motors are used for items delivery. When the can comes out it is detected by a light sensor. The light sensor then stops the spinning motor.   |  | | --- | | untitled.bmp | | **Figure1.18 Simple vending machine.** |  1. Create an I/O assignment list. 2. Analyze the system requirements and draw the Ladder diagram for this control task. 3. Simulate and test your program using the LOGO! Soft comfort software. |